

IN THE SPECIFICATION:

Please amend the first full paragraph appearing on page 2 as follows:

As video data is increasingly used in computer systems in applications such as video conferencing and video recording, computer systems often cannot keep pace with the computational requirements of video data. Video data streams typically have extremely large bandwidth ~~requires~~ requirements that can tax the capabilities of even the most high-speed processor to compress the video data for storage, or for transmission across a computer network or a telephone system. This compression is typically performed by a central processing unit (CPU) in a computer system with a resulting loss in image clarity due to the failure of the CPU to keep pace with the video data. Complex ~~scenes~~, scenes having many elements that are in motion represent the greatest challenge because they place a tremendous burden on the CPU during the compression and data transfer processes.

Please amend the first full paragraph appearing on page 3 as follows:

What is needed is an apparatus or a method for off-loading the ~~time-consuming~~ time-consuming task of computing the difference between successive frames of video data from the CPU of a ~~computing~~ computer.

Please amend the last full paragraph appearing on page 9 as follows:

The circuitry illustrated in FIG. 3 operates as follows. Video data in YUV form 104 from video unit 102 streams into video input buffer 304 through color space conversion module 302. From video input buffer 304, this video data feeds through MUX 312 and I/O buffers 316 into unmodified video data 110 within memory 122. At the same time, data for a previous frame from unmodified video data 110 in memory 122 feeds into previous frame buffer 306 through I/O buffer 316. From previous frame buffer 306, this data feeds into XOR unit 308. XOR unit 308 computes the difference between data from the previous frame, stored in previous frame buffer 306, and data from the current frame, stored in video input buffer 304. The output of XOR unit 308 feeds into result buffer 310. From result buffer ~~319~~, 310, this data feeds through

MUX 312 and I/O buffers 316 into an area for storing XOR video data 112 within memory 112. CPU 120 then uses this difference information to compress the video data.

Please amend the paragraph bridging pages 10 and 11 as follows:

FIG. 4 is a flow chart illustrating a method for compressing video data in a computer system in accordance with an embodiment of the present invention. This flow chart is divided into two columns. The column on the ~~left-hand-side~~ left-hand side represents operations of the computational unit, and the column on the ~~right-hand-side~~ right-hand side represents operations of the memory system. In this embodiment, the system starts in state 400. From state 400, the computational unit proceeds to state 402. In state 402, the computational unit receives a stream of data from a current video frame from a video source. The computational unit next proceeds to state 404. In state 404, the computational unit performs a color space conversion on the video data. The computational unit next proceeds to state 406. In state 406, the computational unit computes a difference frame from a current video frame and a previous video frame received from the memory system "on-the-fly" as the current video frame streams into the computer system. In one embodiment, this difference computation takes place without intervention by the CPU 120. The computational unit next proceeds to state 412. In state 412, the computational unit produces compressed video data using the difference frame. The computational unit then loops back around to state 402 to process more video data.